index

A

Acceptable Use Policy 479
access control lists (ACLs) 189, 193
access control, database
  network access control 310–311
  overview 311
  using IAM service 309–310
account, AWS
  creating
    choosing support plan 26
    contact information 24
    creating key pair 29–32
    login credentials 23
    payment details 24
    signing in 28–29
    verifying identity 25
  security
    authentication roles 177–179
    authorization policies 174–176
    creating users 176–177
    IAM service 173–174
    root user 172–173
ACID (atomicity, consistency, isolation, and durability) 294, 349
ACLs (access control lists) 189, 193
agent forwarding 187–188
AKI (Amazon Kernel Image) 63
alarms, CloudWatch 203, 387–388
Amazon API Gateway 229
Amazon EFS. *See* EFS (Elastic File System)
Amazon ElastiCache. *See* ElastiCache
AMI (Amazon Machine Image) 62–63, 139–140
Apache Bench 479
APIs (application programming interfaces) 102
apt package manager 75
archiving objects 242
ARN (Amazon Resource Name) 175
asynchronous decoupling
  consuming messages 425–428
  converting synchronous process to asynchronous 421–422
  creating SQS queue 423
  overview 420–421
  sending messages to queue 423–425
  SQS messaging limitations 428–429
  URL2PNG application example 422–423
atomic operations 255
atomicity, consistency, isolation, and durability (ACID) 294, 349
auto-scaling 397, 465
  groups 397–398, 466
  overview 396–407
  triggering
    based on CloudWatch metrics 472–474
    based on schedule 471–472
    overview 469–470
automating operational tasks, with AWS Lambda
  adding tag containing owner of EC2 instance automatically 218–226
    authorizing Lambda function to use other AWS services with IAM role 224–226
    deploying Lambda function with SAM 226
    implementing Lambda function in Python 222–223
    setting up Lambda function with SAM 223–224
    subscribing to CloudWatch events 219–222
  building website health check 203–218
    accessing endpoints within VPC 217–218
    creating Lambda function 204–210

automating operational tasks, with AWS Lambda *(continued)*
- monitoring Lambda function with CloudWatch metrics and alarms 212–216
- using CloudWatch to search through Lambda function's logs 210–212

executing code 200–203
- comparing AWS Lambda with virtual machines (Amazon EC2) 202–203
- running code on AWS Lambda 201–202

Free Tier 228
impacts of serverless pricing model 228–229
limitations of 227–228
pricing model 228
use cases
- data processing 230
- IoT back end 231
- web application 229

AWS (Amazon Web Services)
advantages
- automation capabilities 10
- fast-growing platform 10
- platform of services 10
- reducing time to market 11
- reliability 11
- scalability 11
- standards compliance 12
- worldwide deployments 12

alternatives 15–16
costs 275
- billing example 13–15
- Free Tier 13
- overview 12–13
- pay-per-use pricing model 15

services overview 16–18
tools
- blueprints 22
- CLI 20
- Management Console 19
- SDKs 21

use cases
- fault-tolerant systems 8–10
- running Java EE applications 7–8
- running web shop 5–7

AWS Elastic Beanstalk 145
applications in Management Console 149–150
creating application 147
creating environment 147
deleting applications 150–151
deployment 457–461
deployment options comparison 138–139
describing status of installation 147–148
overview 145–146
uploading Zip archive 147

AWS Lambda
adding tag containing owner of EC2 instance automatically 218–226
- authorizing Lambda function to use other AWS services with IAM role 224–226
- deploying Lambda function with SAM 226
- implementing Lambda function in Python 222–223
- setting up Lambda function with SAM 223–224
- subscribing to CloudWatch events 219–222

building website health check with 203–218
- accessing endpoints within VPC 217–218
- creating Lambda function 204–210
- monitoring Lambda function with CloudWatch metrics and alarms 212–216
- use CloudWatch to search through Lambda function's logs 210–212

executing code with 200–203
- comparing AWS Lambda with virtual machines (Amazon EC2) 202–203
- running code on AWS Lambda 201–202

Free Tier 228
impacts of serverless pricing model 228–229
limitations 227–228
pricing model 228
use cases
- data processing 230
- IoT back end 231
- web application 229

AWS OpsWorks
deployment options comparison 138–139
multilayer applications using
- accessing kiwiIRC 162–163
- adding app to Node.js layer 160–161
- adding client and server instances for IRC chat 161
- creating custom layers 156–159
- creating Node.js layer 155–156
- creating stack 154
- deleting stacks 163–164

overview 151–153

AZ (availability zones) 274, 314, 407
defined 392
outages 392–396
- auto-scaling 396–399
- IP addresses and 407–411
- network-attached storage and 402
- recovering failed EC2 instance to another availability zone 399–407

recovering failed EC2 instance to another availability zone 399–407
redundancy 436

B

backup 289–292
  automated snapshots 303–304
  copying database to other region 307
  manual snapshots 304–305
  restoring database 305–307
!Base64 function 142
bastion host
  defined 186
benchmarking 479
block-level storage
  instance stores
    backups 272
    overview 268–271
    performance testing 272
    viewing and mounting volumes 271–272
  network-attached storage
    backing up data 266–268
    creating volumes 260–261
    managing volumes 261–263
    performance improvements 263
BlockDeviceMappings 269–270
blueprints
  AWSTemplateFormatVersion value 123
  example template 127–133
  Lambda function configuration 205
  outputs structure 126
  overview 22, 121–122
  parameters structure 123–125
  resources structure 125–126
bucket policies 253
BucketNotEmpty error 240
buckets, S3
  configuring 253–254
  creating programmatically 253
  linking custom domain 255
  listing files 251–252
  setting up 249
  versioning 239–240
burst rate, EFS 286
burstable performance 474
BurstCreditBalance metric 286–287

C

Cache Engine 334
cache, invalidating 439
caching data in memory. *See* ElastiCache
calculator for monthly costs 13
Cassandra 354
CDN (content delivery network) 6, 83, 252
Chef 138, 152
CIDR (Classless Inter-Domain Routing) 185
CLI (command-line interface) 237
  –query option 114–115
  advantages of using scripts 108–109
  configuring user authentication 110–113
  creating virtual machine using script 113–115
  help keyword 113
  installing
    on Linux 109
    on Mac OS 109
    on Windows 109–110
  listing EC2 instances 113
  overview 20
  usage overview 113
cloud computing
  deployment environment 136–137
  overview 4–5
CloudFormation
  alternatives 133
  blueprints
    AWSTemplateFormatVersion value 123
    example template 127–133
    outputs structure 126
    overview 121–122
    parameters structure 123–125
    resources structure 125–126
CloudFormation template
  defining Lamda function with SAM within 224
  minimal 327–328
CloudWatch
  creating alarm 387–388
  monitoring Lambda function with metrics and alarms 212–216
  overview 385–386
  subscribing to events 219–222
  triggering auto-scaling based on metric 472–474
CNAME records 255
cold-start 227
content delivery network (CDN) 6, 83, 252–253
cookbooks, Chef 152
cost
  AWS Lamda 203
  billing example 13–15
  DynamoDB service 352
  Free Tier 13
  optimizing for virtual machines
    overview 92
    reserved virtual machines 93–94
    spot instances 95–100
  overview 12–13
  pay-per-use pricing model 15
CPU load 474
CPUUtilization metric 344–345
CREATE_COMPLETE state 43, 298–299, 479
CREATE_IN_PROGRESS state 43
CRUD (create, remove, update, delete) 248

D

data centers
  locations 4, 12
  starting virtual machine in different 82–86
data processing, automating operational tasks with AWS Lambda 230
databases
  for WordPress blogs example 49–51
  network access control 310–311
dd utility 263
DDoS (Distributed Denial of Service) attacks 167
declarative approach, defined 121
decoupling
  asynchronous, with message queues
    consuming messages 425–428
    converting synchronous process to asynchronous 421–422
    creating SQS queue 423
    overview 420–421
    sending messages to queue 423–425
    SQS messaging limitations 428–429
    URL2PNG application example 422–423
  concept explained 413–415
  dynamic EC2 instance pools
    by load balancer 476–480
    by queue 481–484
    overview 475–482
  redundant EC2 instances 436–437
  synchronous, with load balancers
    overview 415–416
    setting up load balancer 416–420
default region setting 39
dependencies, CloudFormation templates 122
deployment
  AWS Lambda 203
  comparison of options 138–139
  defined 135
  in scalable cloud environment 136–137
  multilayer applications with AWS OpsWorks Stacks 152–153
  running script on virtual machine startup
    overview 139
    using user data 140
  web applications with AWS Elastic Beanstalk
    components 146
    creating application 147
    creating environment 147
    deleting applications 150–151
    uploading Zip archive 147
DevOps (development operations) movement 104–105
df command 262
disaster recovery 411–412
Discourse application, installing with CloudFormation 336–344
  cache 338–339
  database 339–340
  testing CloudFormation template 342–344
  virtual machine 340–342
  VPC 337–338
Distributed Denial of Service (DDoS) attacks 167
DLQ (dead-letter queue) 456
DNS (Domain Name System) 83
DSL (domain-specific language) 152
DSS (data security standard) 12
dynamic EC2 instance pools
  decoupling
    by load balancer 476–480
    by queue 481–484
    overview 475–482
  managing 465–469
DynamoDB service
  administrative tasks 352
  costs 352
  deleting data 373–374
  eventual consistency 372–373
  modifying data 374–375
  NoSQL comparison 354
  overview 349–351
  primary key
    defined 355
    partition key and sort key 355–356
  querying data
    by key 365–366
    by key and filter 366–368
    overview 364–365
    retrieving all items 371–372
    using secondary indexes 368–371
  RDS vs. 353
  running DynamoDB locally 356
  scaling 375–379
  tables
    creating 358
    overview 354–355
    using partition key 358–359
    using partition key and sort key 360–361

E

EBS (Elastic Block Store)
  backing up data from 266–268
  creating volumes 260–261
  defined 432
  managing volumes 261–263
  performance improvements 263
EBS. *See* instance stores

EC2 (Elastic Compute Cloud) service
  comparing AWS Lambda with 202–203
  defined 3, 37
  recovering instances 386
  See also virtual machines
EFS (Elastic File System)
  backing up data 289–292
    using CloudFormation to describe EBS volume 290
    using EBS volume 290–292
  burst rate 286
  filesystem, creating 277–278
    pricing 277–278
    using CloudFormation to describe filesystem 277
  filesystem, monitoring 286–289
    Max I/O Performance Mode, whether to use 286
    monitoring permitted throughput 287–288
    monitoring usage 288–289
  mount target, creating 278–279
  mounting EFS share on EC2 instances 280–283
  performance, tweaking 284–286
    expected throughput 285–286
    Performance mode 285
  sharing files between EC2 instances 283–284
Elastic Beanstalk. *See* AWS Elastic Beanstalk
elastic IP addresses 86
ElastiCache
  access to cache, controlling 334–336
    access to configuration 334
    cluster and data access 335–336
    network access 334–335
  cache deployment options 330–334
    Memcached 330–331
    Redis 331–334
  creating cache cluster 327–330
    minimal CloudFormation template 327–328
    testing Redis cluster 328–330
  monitoring cache 344–346
    monitoring host-level metrics 344–345
    Redis replication, whether up-to-date 345–346
    sufficient memory 345
  performance of cache, tweaking 346–348
    compressing data 348
    selecting right cache node type 347
    selecting right deployment option 347–348
elasticity 464
ELB (Elastic Load Balancing) 37, 415
ENI (Elastic Network Interface) 432
environment variables
  temporary credentials and 224
  to pass settings to Lambda function 209
environment, defined 458
Erlang programming language 437
Errors metric, CloudWatch 212
eventual consistency 255, 372–373, 439
Evictions metric 344–345

F

failure recovery
  availability zone outages
    auto-scaling 396–399
    availability zones 392–396
    IP addresses and 407–411
    network-attached storage and 402
    recovering failed EC2 instance to another availability zone 399–407
fault-tolerance
  code considerations
    “let it crash” concept 437–438
    idempotent retry 438–439
  defined 432
  high availability vs. 384
  overview 431–433
  redundant EC2 instances
    decoupling required 436–437
    overview 434
    removing single point of failure 434–436
  web application
    creating process 446–448
    idempotent state machine 443–444
    idempotent state transitions 444
    Imagery application overview 440–461
    looking up process 448–449
    server 445–446
    uploading files 449–452
    worker for consuming SQS messages 452–455
  web application deployment
    DynamoDB 455–457
    Elastic Beanstalk for worker 459–461
    IAM roles 457
    overview 455
    S3 455–457
    SQS 455–457
fdisk utility 261, 271
filesystem. *See* EFS (Elastic File System)
  See also EBS; instance stores
firewalls 69, 158
Free Tier 13
fsfreeze command 267

G

generations, defined 65
GiB (gibibyte) 265

Glacier service
  adding lifecycle rule to bucket 242
  moving objects to and from 245–248
  S3 service 241
globally unique identifiers 236–237

H

HA (high availability)
  defined 384, 432
  disaster-recovery requirements 411–412
  fault tolerance vs. 384
  for databases 311–314
  recovering from availability zone outages
    auto-scaling 396–399
    availability zones 392–396
    IP addresses and 407–411
    network-attached storage and 402
    recovering failed EC2 instance to another availability zone 399–407
  recovering from EC2 instance failure with CloudWatch
    creating alarm 387–388
    overview 385–386
  redundant EC2 instances for
    decoupling required for 436–437
    overview 434
    removing single point of failure 434–436
host machines 60
host-level metrics 344–345
httpd-tools package 479
HVM (Hardware Virtual Machine) 64

I

IaaS (infrastructure as a service) 5, 104
IAM (Identity and Access Management) service 173–174, 309, 334
  database access control using 309–310
  roles
    authorizing Lambda function to use AWS services with 224–226
    deployment 457
ICMP (Internet Control Message Protocol) 182–183
idempotent
  defined 429, 438
  retry 438–439
  state machine 443–444
  state transitions 444
IGW (internet gateway) 190–191
infrastructure as code
  CLI
    advantages of using scripts 108–109
    configuring user authentication 110–113
    creating virtual machine using script 113–115
    help keyword 113
    installing 109–110
    listing EC2 instances 113
    usage overview 113
  defined 104
  DevOps movement 104–105
  JIML 105–108
  using SDKs 117
inline policy 176
installation
  software on virtual machines 75
instance family groups 65
instance stores
  backups and 272
  overview 268–271
  performance testing 272
  viewing and mounting volumes 271–272
instances, defined 152
IOPS (input/output operations per second) 264–266
IoT 231
IP (Internet Protocol) 180
IP addresses
  allocating for virtual machine 86–88
  public vs. private 184
IRC (Internet Relay Chat) 153

J

Java EE applications 7–8
JIML (JSON Infrastructure Markup Language) 105–108
JMESPath 114
jump boxes 186

K

key pair for SSH
  creating 29–32
  selecting for virtual machine 70–72
key-value stores 325, 350, 364
kiwiIRC 154–161, 163

L

Lambda function 203
  configuring scheduled event triggering 208
  creating 205–206
  defining with SAM within CloudFormation template 224
  invocations 210
  log messages from 211

Lambda function *(continued)*
  monitoring with CloudWatch metrics and alarms 212
  website health check and environment variables to pass settings to 209
Lambda. *See* AWS Lambda
launch configurations 397
lazy-loading 322
"let it crash" concept 437–438
lifecycle rules 242
linkchecker tool 75
Linux
  connecting to virtual machines from 73
  installing CLI on 109
load balancers
  decoupling dynamic EC2 instance pools by 476–480
  for WordPress blogs example 47–48
  synchronous decoupling with
    overview 415–416
    setting up load balancer 416–420
load monitoring 77–78
load tests 479
logs
  viewing for AWS Elastic Beanstalk application 150
  viewing for virtual machines 76–77

M

Mac OS X
  connecting to virtual machines from 73
  installing CLI on 109
  key file permissions 31
managed policy 176
Management Console
  overview 19
  signing in 28
Max I/O Performance Mode 286
Memcached 326, 330–331
memory, caching data in. *See* ElastiCache
message queues
  consuming messages 425–428
  converting synchronous process to asynchronous 421–422
  creating SQS queue 423
  overview 420–421
  sending messages to queue 423–425
  SQS messaging limitations 428–429
  URL2PNG application example 422–423
metadata 236
MFA (multifactor authentication) 172
mkfs command 262
MongoDB 354
monitoring
  load 77–78
  viewing logs 76–77
MySQL databases
  costs 300
  database instance information 299–300
  exporting 301
  for WordPress blogs example 49–51
  WordPress platform using 297–299

N

Name tag 67
NAT (Network Address Translation) 7, 184, 190, 196–197
Neo4j 354
network access control 310–311
network-attached storage
  availability zone outages and 402
  backing up data from 266–268
  creating volumes 260–261
  managing volumes 261–263
  performance improvements 263
networking
  ACLs (access control lists) 189, 193
  controlling traffic 183–189
    overview 179–181
    using security groups 181–182
  for virtual machines
    allocating fixed public IP address 86–88
    creating additional network interface 88–92
  using access control lists (ACLs) 189, 193
Node.js
  installing 117, 249, 361, 423
NoSQL databases 229
  RDS vs. DynamoDB 353
  running DynamoDB locally 356
  scaling 375–379
  tables
    creating 358
    overview 354–355
    using partition key 358–359
    using partition key and sort key 360–361

O

object stores 229
  backing up data using 238–240
  concepts of 236–237
  data consistency 255
  Glacier service
    adding lifecycle rule to bucket 242
    creating S3 bucket for 241–242

object stores *(continued)*
    moving objects to and from 245–248
    S3 service vs. 241
  S3 service 237
  selecting keys for objects 256–257
  static web hosting using
    accessing website 254–255
    configuring bucket 253–254
    creating bucket for 253
  storing objects programmatically
    installing web application 249–250
    listing files in bucket 251–252
    overview 248–249
    setting up S3 bucket 249
    uploading files to S3 250–251
on-demand instances 92
OpenSwan VPN server
  installing VPN with script 144–145
  overview 140–142
optimistic locking 447–448
OS (operating system) 46, 62–64

P

PaaS (platform as a service) 5
partition key 355–356, 358–361
PCI (payment card industry) 12
performance
  database
    increasing database resources 314–315
    using read replication 316–318
  increasing speed using CDN 252–253
PermittedThroughput metric 286–287
pessimistic locking 448
pip tool 109
policies, authorization 174–176
PostgreSQL 339
primary key, DynamoDB
  defined 355
  partition key 355–356
private IP addresses 184
public IP addresses 86–88, 184
putItem operation 362
PuTTY 32, 188
Python
  implementing Lambda function 222–223
  installing packages 109

Q

querying data, from DynamoDB service
  by key 365–366
  by key and filte 366–368
  overview 364–365
  retrieving all items 371–372
  using secondary indexes 368–371
queue
  decoupling dynamic EC2 instances pools
    by 481–484
queuing theory 345

R

RDP (Remote Desktop Protocol) 29
RDS (relational database service)
  access control
    network access control 310–311
    overview 311
    using IAM service 309–310
  backup/restore
    automated snapshots 303–304
    copying database to other region 307
    costs 308
    manual snapshots 304–305
    restoring database 305–307
  defined 37
  DynamoDB service vs. 353
  failures possible for 432
  high availability 311–314
  importing data 300–303
  monitoring database 318–320
  MySQL databases
    costs 300
    database instance information 299–300
    WordPress platform using 297–299
  overview 294–296
  performance
    increasing database resources 314–315
    using read replication 316–318
Redis
  cluster with cluster mode disabled 332
  cluster with cluster mode enabled 332–334
  replication, whether up-to-date 345–346
  single-node cluster 331–332
  SortedSet 326
  testing cluster 328–330
regions 39, 392
reliability, of AWS 11
reserved instances 92–94
resource groups 44
REST API 103–104
roles, authentication 177–179
root user 172–174
RPO (recovery point objective) 411–412
RTO (recovery time objective) 411–412

S

S3 (Simple Storage Service)
  backing up data using 238–240
  data consistency 255
  defined 3
  deployment 455–457
  Glacier service 241
  linking custom domain to bucket 255
  overview 237
  selecting keys for objects 256–257
  static web hosting using
    accessing website 254–255
    configuring bucket 253–254
    creating bucket for 253
  storing objects programmatically
    installing web application 249–250
    listing files in bucket 251–252
    overview 248–249
    setting up S3 bucket 249
    uploading files to S3 250–251
  versioning for 239–240
SaaS (software as a service) 5
SAM (Serverless Application Model)
  defining Lambda function with 224
  deploying Lambda function with 226
  setting up Lambda function with 223–224
scaling
  advantages of AWS 11
  based on CPU load 474
  decoupling dynamic EC2 instance pool
    by load balancer 476–480
    by queue 481–484
    overview 475–482
  DynamoDB service 375–379
  general discussion 463–465
  managing dynamic EC2 instance pool 465–469
  policies 465
  triggering auto-scaling
    based on CloudWatch metrics 472–474
    based on schedule 471–472
    overview 469–470
scan operation 371–372
schedule expressions 207
scheduled events 203, 208
SDKs (software development kits)
  overview 21
  platform and language support 117
secondary indexes 368–371
security
  AWS account
    authentication roles 177–179
    authorization policies 174–176
    creating users 176–177
    IAM service 173–174
    importance of securing 171–177
    root user 172–173
  controlling network traffic
    allowing ICMP traffic 182–183
    allowing SSH traffic 183–184
    allowing SSH traffic from IP address 184–185
    allowing SSH traffic from security group 185–189
    overview 179–181
    using security groups 181–182
  creating VPC
    accessing internet via NAT gateway 196–197
    adding private Apache web server subnet 194–195
    creating IGW 190–191
    defining public bastion host subnet 192–193
    launching virtual machines in subnets 195–196
    overview 189–190
  shared responsibility with AWS 167
  updating software
    checking for security updates 168–169
    installing updates on running virtual machines 170–171
    installing updates on startup 169–170
security groups 338–342
  allowing SSH traffic 185–189
  defined 38
  overview 181–182
sharing data volumes between machines. *See* EFS (Elastic File System)
single point of failure (SPOF) 332
single-node cluster, Redis 331–332
snapshots, database
  automated 303–304
  copying automated as manual 305
  manual 304–305
software, installing on virtual machines 75
SPOF (single point of failure) 332, 432, 434–436
spot instances 92, 95–100
SQS (Simple Queue Service)
  consuming messages 425–428
  creating queue 423
  creating worker for consuming messages 452–455
  defined 415
  deployment 455–457
  limitations of 428–429
  sending messages to queue 423–425
SSH traffic, allowing 183–184
  from IP address 184–185
  from security group 185–189
stacks 128, 152
stateless servers 249

statelessness 465
static web hosting
  accessing website 254–255
  configuring bucket 253–254
  creating bucket 253
stopping vs. terminating machines 78
storage
  instance stores
    backups 272
    overview 268–271
    performance testing 272
    viewing and mounting volumes 271–272
  network-attached storage
    backing up data 266–268
    creating volumes 260–261
    managing volumes 261–263
    performance improvements 263
streams, DynamoDB 364–365
strongly consistent reads 373, 375
subnet group 338–340
subnets 436
synchronous decoupling
  overview 415–416
  setting up load balancer 416–420
system status checks 385

T

tables, DynamoDB
  creating 358
  overview 354–355
  using partition key 358–359
  using partition key and sort key 360–361
tags, defined 67
templates, CloudFormation
  AWSTemplateFormatVersion value 123
  example of 127–133
  outputs structure 126
  overview 121–122
  parameters structure 123–125
  resources structure 125–126
terminating vs. stopping machines 78
Terraform 133
TiB (tebibyte) 265
tools
  blueprints 22
  CLI 20
  Management Console 19
  SDKs 21
Troposphere 133
TTL (time to live) 415

U

updates, security
  checking for 168–169
  installing on running virtual machines 170–171
  installing on startup 169–170
use cases
  fault-tolerant systems 8–10
  running Java EE applications 7–8
  running web shop 5–7
user data 140
users, creating 176–177
UUID (universally unique identifier) 438

V

versioning
  for applications 146
  for S3 buckets 239–240
  versions defined 457
virtual appliances 63–64
virtual machines 37
  allocating fixed public IP address 86–88
  changing size of 79–82
  comparing AWS Lambda with 202–203
  connecting to
    from Linux 73
    from Mac OS X 73
    from Windows 73
    login message when connecting 74–75
    overview 72–75
  cost optimization
    overview 92
    reserved virtual machines 93–94
    spot instances 95–100
  creating additional network interface for 88–92
  creating using CLI script 113–115
  installing software on 75
  launching
    choosing size of 64–65
    naming 66
    overview 60–61
    selecting key pair for SSH 70–72
    selecting OS 62–64
  listing EC2 instances using CLI 113
  monitoring
    load 77–78
    viewing logs 76–77
  overview 60–75
  running script on virtual machine startup
    application update process 145
    overview 139
    using user data 140

virtual machines *(continued)*
security updates for
checking 168–169
installing on running virtual machines 170–171
installing on startup 169–170
shutting down 78–79
starting in another data center 82–86
virtualization, granularity of 202
VMs (virtual machines) 7
See also EC2 Instance
VPC (Virtual Private Cloud) 337–338, 386, 395
accessing endpoints within 217–218
accessing internet via NAT gateway 196–197
adding private Apache web server subnet 194–195
creating IGW 190–191
defined 432
defining public bastion host subnet 192–193
launching virtual machines in subnets 195–196
overview 189–190
VPN (Virtual Private Network) 7
installing VPN with script 144–145
overview 140–142
using CloudFormation to start virtual machine with user data 142–144

W

web applications
using AWS Elastic Beanstalk
components of 146
creating application 147
website health check
building 203–218
wildcard character (*) 174
Windows
connecting to virtual machines from 73
EC instances on 270
installing CLI on 109–110
SSH client on 32
WordPress
AWS installation example
costs 52–53
creating infrastructure 37–44
load balancer 47–48
MySQL database 49–51
web servers 45–47

X

Xen 64

Y

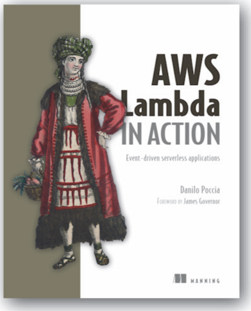
yum package manager 85–86, 144, 168

RELATED MANNING TITLES

Learn Amazon Web Services in a Month of Lunches

by David Clinton

ISBN: 9781617294440
328 pages, $39.99
August 2017

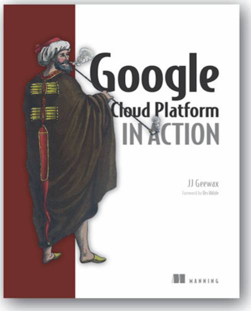

AWS
Lambda
IN ACTION
Event-driven serverless applications
Danilo Poccia
Foreword by James Governor
MANNING

AWS Lambda in Action

Event-driven serverless applications

by Danilo Poccia

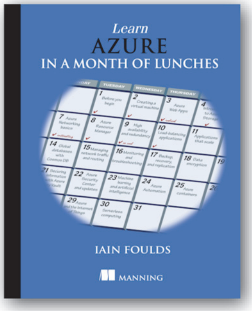
ISBN: 9781617293719
384 pages, $49.99
November 2016



Google Cloud Platform in Action

by JJ Geewax

ISBN: 9781617293528
632 pages, $59.99
September 2018



Learn Azure in a Month of Lunches

by Iain Foulds

ISBN: 9781617295171
375 pages, $44.99
September 2018

For ordering information go to www.manning.com

Storage & Database

Abbr.	Name	Description	Where
S3	Amazon Simple Storage Service	Object store to save data without any size restrictions	8
	Amazon Glacier	Inexpensive data archive solution	8.4
EBS	Amazon Elastic Block Store	Network attached block level storage for EC2 instances	9.1
	Amazon EC2 Instance Store	Block level storage for EC2 instances	9.2
EFS	Amazon Elastic File System	Scalable network file system based on NFSv4	10
RDS	Amazon Relational Database Service	MySQL, Oracle Database, Microsoft SQL Server, or PostgreSQL	11
	Amazon DynamoDB	Proprietary NoSQL key-value database with limited but powerful options for queries	13
	Amazon ElastiCache	In-memory data store based on Redis or Memcached typically used to cache data	12

Architecting on AWS

Abbr.	Name	Description	Where
AZ	Availability Zone	Group of isolated data centers within a region	14.2
ASG	Amazon EC2 auto-scaling group	Observes a fleet of servers: Replaces faulty servers and increases/decreases the size of the fleet based on external triggers like CPU usage	17
	Amazon CloudWatch	Keeps track of metrics and triggers alarms when a threshold is reached	14.1
ELB	Elastic Load Balancing	Load balancer for EC2 instances	15.1
ALB	Application Load Balancer	Layer 7 load balancer with HTTP and HTTPS support	15.1
SQS	Amazon Simple Queue Service	Message queue	15.2

CLOUD/AWS

Amazon Web Services IN ACTION Second Edition

Michael Wittig • Andreas Wittig

The largest and most mature of the cloud platforms, AWS offers over 100 prebuilt services, practically limitless compute resources, bottomless secure storage, as well as top-notch automation capabilities. This book shows you how to develop, host, and manage applications on AWS.

Amazon Web Services in Action, Second Edition is a comprehensive introduction to deploying web applications in the AWS cloud. You'll find clear, relevant coverage of all essential AWS services, with a focus on automation, security, high availability, and scalability. This thoroughly revised edition covers the latest additions to AWS, including serverless infrastructure with AWS Lambda, sharing data with EFS, and in-memory storage with ElastiCache.

What's Inside

- Completely revised bestseller!
- Secure and scale distributed applications
- Deploy applications on AWS
- Design for failure to achieve high availability
- Automate your infrastructure

Written for mid-level developers and DevOps engineers.

Andreas and Michael Wittig are software engineers and DevOps consultants focused on AWS. Together, they migrated the first bank in Germany to AWS in 2013.

To download their free eBook in PDF, ePub, and Kindle formats,
owners of this book should visit
manning.com/books/amazon-web-services-in-action-second-edition

> "Slices through the complexity of AWS using examples and visuals to cement knowledge in the minds of readers."
>
> —From the Foreword by
> Ben Whaley
> AWS community hero and author

> "The authors' ability to explain complex concepts is the real strength of the book."
>
> —Antonio Pessolano
> Consoft Sistemi

> "Useful examples, figures, and sources to help you learn efficiently."
>
> —Christof Marte, Daimler-Benz

> "Does a great job of explaining some of the key services in plain English so you have the knowledge necessary to dig deeper."
>
> —Ryan Burrows
> Rooster Park Consulting

Free eBook
See first page

ISBN-13: 978-1-61729-511-9
ISBN-10: 1-61729-511-6
55499
9 781617 295119

MANNING $54.99 / Can $72.99 [INCLUDING eBOOK]